

# OPERATING-CHARACTERISTIC GUIDED DESIGN OF GROUP-SEQUENTIAL TRIALS

ANIKO SZABO

**ABSTRACT.** Group-sequential designs are commonly used for clinical trials to allow early stopping for efficacy or futility. While the design of a single-stage randomized trial is guided by a target power for an alternative hypothesis of interest, the addition of interim analyses is driven by technical choices that are less understandable for clinicians. For example, the commonly used Lan-DeMets methodology requires specification of the timing of analyses and error spending functions. Since the rationale and effect of these technical choices is often unclear, the operating characteristics of the final design are explored under various values of the parameter of interest, and the design is then adjusted until desired properties are obtained.

In this work we develop methods for constructing designs that achieve the desired operating characteristics without the need to specify error spending functions or the timing of analyses. Specifically, we consider designing a study for the mean difference  $\theta$  of a normally distributed outcome with known variance. The null hypothesis  $H_0 : \theta = \theta_0$  is tested versus  $H_A : \theta = \theta_A$ , with power  $\pi$  at a significance level  $\alpha$ . The interim analyses are designed so that for a pre-specified sequence  $\theta_{Ak}$  the study stops for efficacy at stage  $k$  with probability  $\pi$  if  $\theta = \theta_{Ak}$ . If stopping for futility is also considered, then the requirement to stop for futility at stage  $k$  with probability  $\pi_F$  if  $\theta = \theta_{0k}$  for pre-specified sequence  $\theta_{0k}$  can also be added. We show that under some monotonicity restrictions, such designs exist for any choice of the timing of interim analyses. Specific designs can be selected by imposing additional optimality requirements, such as minimizing the expected sample size under the target alternative  $\theta_A$ , or the average sample size under a weighted selection of the alternatives.

## 1. SETUP AND NOTATIONS

All the calculations are set up assuming a setting with a normally distributed outcome with a known variance, but many common settings can be reduced to / approximated by this special case.

Let  $X_i \sim N(\theta, \sigma^2)$  be a sequence of independent identically distributed measurements. We are designing a study to test the following hypotheses:

$$H_0 : \theta = \theta_0 \quad \text{versus} \quad H_A : \theta \geq \theta_0 \quad (1)$$

with type I error  $\alpha$  and power  $\pi$  at a pre-specified value  $\theta = \theta_A$ .

The test statistic for  $H_0$  based on  $n$  values is  $Z(n) = (\bar{X}_n - \theta_0) \sqrt{n} / \sigma = \sqrt{\mathcal{I}_n}(\bar{X}_n - \theta_0)$ , where  $\mathcal{I}_n = [\sigma^2/n]^{-1}$  is the Fisher information after  $n$  observations. Then for any value of  $\theta$

$$Z(n) \sim N(\sqrt{\mathcal{I}_n}(\theta - \theta_0), 1), \quad (2)$$

and for  $n_1 \leq n_2$

$$\text{Cov}(Z_{n_1}, Z_{n_2}) = \sqrt{\mathcal{I}_{n_1} / \mathcal{I}_{n_2}}. \quad (3)$$

This information-based formulation holds (approximately) for a variety of potential designs and outcomes, including two-arm randomized and cross-over studies with normal or binary outcomes.

The study will use a group-sequential design with  $K$  analyses, conducted at relative times  $t_1, \dots, t_K = 1$ . The  $k^{\text{th}}$  analysis will be conducted after  $n_k = N \times t_k$  observations have been obtained, where  $N$  is the maximum sample size of the study. Based on the test statistic  $Z_k = Z(n_k)$  at analysis  $k = 1, \dots, K-1$ , the trial can be stopped early for efficacy (reject  $H_0$ ) if  $Z_k \geq u_k$ , stopped early for futility (fail to reject  $H_0$ ) if  $Z_k \leq l_k$ , or continued to the next stage. At the final,  $K^{\text{th}}$  analysis,  $H_0$  would either be rejected if  $Z_K \geq u_K = l_K$ , or we would fail to reject  $H_0$  otherwise. As a special case, setting  $l_k = -\infty$  will result in a study when interim stopping for futility is not considered.

The decision boundaries  $u_k, l_k, k = 1, \dots, K$ , and the maximum information target ( $\sim$  sample size)  $\mathcal{I}_{\max}$  have to be selected so that the overall type I error and power of the study are maintained at their design

TABLE 1. Notations

Characteristic	Notation	Event
<b>No or non-binding futility boundary</b>		
Rejection <i>at stage k</i>	$\mathcal{R}_k$	$\{Z_1 < u_1, \dots, Z_{k-1} < u_{k-1}, Z_k \geq u_k\}$
Continuation <i>at stage k</i>	$\mathcal{C}_k$	$\{Z_1 < u_1, \dots, Z_{i-1} < u_{k-1}, Z_k < u_k\}$
<b>Non-binding futility boundary</b>		
Acceptance <i>at stage k</i>	$\mathcal{A}_k$	$\{l_1 < Z_1 < u_1, \dots, l_{k-1} < Z_{k-1} < u_{k-1}, Z_i \leq l_k\}$
<b>Binding futility boundary</b>		
Rejection <i>at stage k</i>	$\mathcal{R}_k^*$	$\{l_1 < Z_1 < u_1, \dots, l_{k-1} < Z_{k-1} < u_{k-1}, Z_k \geq u_k\}$
Continuation <i>at stage k</i>	$\mathcal{C}_k^*$	$\{l_1 < Z_1 < u_1, \dots, l_{k-1} < Z_{k-1} < u_{k-1}, l_k < Z_k < u_k\}$
Acceptance <i>at stage k</i>	$\mathcal{A}_k$	$\{l_1 < Z_1 < u_1, \dots, l_{k-1} < Z_{k-1} < u_{k-1}, Z_i \leq l_k\}$

TABLE 2. Target operating characteristics

Characteristic	Event	$\theta$	Probability target
<b>No or non-binding futility boundary</b>			
Overall type I error	$\bigcup_{i=1}^K \mathcal{R}_i$	$\theta_0$	$\leq \alpha$
Overall power	$\bigcup_{i=1}^K \mathcal{R}_i$	$\theta_A$	$\geq \pi$
Stop by stage $k$ for efficacy	$\bigcup_{i=1}^k \mathcal{R}_i$	$\theta_{Ak}$	$\geq \pi_E$
<b>Non-binding futility boundary</b>			
Stop by stage $k$ for futility	$\bigcup_{i=1}^k \mathcal{A}_i$	$\theta_{0k}$	$\geq \pi_F$
<b>Binding futility boundary</b>			
Overall type I error	$\bigcup_{i=1}^K \mathcal{R}_i^*$	$\theta_0$	$\leq \alpha$
Overall power	$\bigcup_{i=1}^K \mathcal{R}_i^*$	$\theta_A$	$\geq \pi$
Stop by stage $k$ for efficacy	$\bigcup_{i=1}^k \mathcal{R}_i^*$	$\theta_{Ak}$	$\geq \pi_E$
Stop by stage $k$ for futility	$\bigcup_{i=1}^k \mathcal{A}_i$	$\theta_{0k}$	$\geq \pi_F$

values, but there are many valid choices. In this *operating characteristic driven* approach we propose to select the boundaries based on the cumulative probabilities of stopping for efficacy or futility at each stage.

**Theorem 1.** *A design satisfying the operating characteristic requirements of Table 2 exists, if the following conditions are satisfied:*

- C1.  $\pi_E \leq \pi$
- C2.  $\theta_{A1} \geq \theta_{A2} \geq \dots \geq \theta_{A,K-1} \geq \theta_A$

*Additionally, if a futility boundary is needed:*

- C3.  $\pi_F \leq 1 - \alpha$
- C4.  $\theta_{01} \leq \theta_{02} \leq \dots \leq \theta_{0,K-1} \leq \theta_0$

*Proof.* We will consider the cases of no, non-binding, and binding futility boundaries separately.

**I. no futility boundary** We will use proof by induction over the number of stages  $K$ . When  $K = 1$ , only the overall type I error and power need to be considered, and the well-known single-stage design achieving the overall information

$$\mathcal{I}_{\text{fix}} = \frac{(z_\alpha + z_{1-\beta})^2}{(\theta_A - \theta_0)^2}$$

will satisfy the requirements with  $u_K = z_\alpha$ .

Now suppose we can construct the desired design for  $K - 1$  stages, and we try to add an additional stage. For the first  $K - 1$  stages select the boundary  $u_1, \dots, u_{K-1}$  and information times  $\mathcal{I}_{n_1}, \dots, \mathcal{I}_{n_{K-1}}$  to achieve over these  $K - 1$  stages stage-specific stopping probabilities  $\pi_E$ , overall power  $\pi_E$ , and overall type I error  $\alpha_{K-1} = \alpha - \Delta\alpha_K$ , where  $0 < \Delta\alpha_K < \alpha$  is an arbitrary value. With these choices, the requirements for all the stage-specific probabilities for the  $K$ -stage design are satisfied. We need to select  $I_{n_K}$  and  $u_K$  to satisfy the overall power and type I error restrictions.

Consider the function  $a(u | \mathcal{I}_N) = Pr(\{\bigcup_{i=1}^{K-1} \mathcal{R}_i\} \cup \{Z_1 < u_1, \dots, Z_{K-1} < u_{K-1}, Z(N) \geq u\} | \theta_0)$ , that gives the type I error if the boundary is set at  $u$  for the last stage for any given  $I_N > I_{n_{K-1}}$ . Under  $H_0$ ,  $Z(N) \sim N(0, 1)$ , so  $a(z_\alpha | \mathcal{I}_N) \geq \alpha$ . On the other hand,  $a(\infty | \mathcal{I}_N) = \alpha - \Delta\alpha_K < \alpha$  by the design of the first  $K - 1$  stages. Since  $a$  is monotone in  $u$ , for any  $I_N$  there exists a cutoff  $u_K(\mathcal{I}_N)$  for which  $a(u_K(\mathcal{I}_N)) = \alpha$ , ie for which the overall type I error is maintained at the desired level.

Next consider the function  $b(\mathcal{I}_N) = Pr(\{\bigcup_{i=1}^{K-1} \mathcal{R}_i\} \cup \{Z_1 < u_1, \dots, Z_{K-1} < u_{K-1}, Z(N) \geq u_K(\mathcal{I}_N) | \theta_A)$  that gives the power to reject  $H_0$  if the final analysis is set at information  $\mathcal{I}_N > I_{n_{K-1}}$ . Since  $\theta_A \leq \theta_{A,K-1}$ ,  $b(\mathcal{I}_{K-1}) \leq \pi_E \leq \pi$ . On the other hand,  $b(\infty) = 1$ , and  $b$  is monotone in  $\mathcal{I}_N$ . Thus we can select a value  $n_K$  such that  $b(\mathcal{I}_{n_K}) = \pi$ , which will provide the desired design.

**II. Non-binding futility boundary** The overall power, type I error, and stage-specific efficacy power requirements do not depend on a non-binding futility boundary, so we can start with a design constructed without a futility boundary. We then need to calculate the lower bounds  $l_k$  to satisfy the futility stopping probabilities. For stage 1, we need  $Pr(Z_1 < l_1 | \theta_{01}) = \pi_F$ , where  $Z_1 \sim N(\sqrt{I_1}(\theta_{01} - \theta_0), 1)$ . Thus  $l_1 = z_{1-\pi_F} + \sqrt{I_1}(\theta_{01} - \theta_0)$  satisfies the target power. Given conditions C3-C4,  $l_1 \leq z_\alpha$ , while  $u_1 \geq z_\alpha$ , so  $l_1 \leq u_1$ .

At stage  $k$ , consider the function  $c(l) = Pr(l_1 < Z_1 < u_1, \dots, l_{k-1} < Z_{k-1} < u_{k-1}, Z_k < l | \theta_{0k})$ . Since  $\theta_{0k} \leq \theta_0$ ,  $c(z_\alpha) \geq 1 - \alpha \geq \pi_F$ , while  $c(-\infty) = 0$ , so a value  $l_k$  can be selected for which  $c(l_k) = \pi_F$ .

**III. Binding futility boundary** The construction follows the induction-based approach of part I. During the inductive step in addition to  $I_K$  and  $u_K$ , we need to find a value for  $l_{K-1}$  that will satisfy the futility boundary restriction. This can be done as shown in part II before proceeding with the selection of the parameters of the last stage. For that step, the definitions of  $a(\cdot)$  and  $b(\cdot)$  have to be slightly adjusted to include the condition of not having exceeded the lower boundary in the previous steps.

□

In the proof of the theorem we have obtained the following uniqueness result:

**Corollary 1.** *Under the assumptions of Theorem ??, an alpha-spending sequence  $\Delta\alpha_k > 0$ ,  $k = 1, \dots, K$  such that  $\sum_{k=1}^K \Delta\alpha_k = \alpha$  uniquely determines a design with the required operating characteristics.*



## 2. MAIN FUNCTION: CALCULATE DESIGN

```
"../R/gsDesignOC.R" 5≡
```

```
#'Find optimal group-sequential design
#
#The \code{gsDesignOC} function finds the analysis times, boundaries, and sample size
#for a group-sequential trial
#
#@export
#param thA numeric; effect size under the alternative hypothesis
#param thA.seq monotone numeric vector of values getting closer to thA from outside the
#th0-thA range (ie, if thA > th0, they should form a decreasing sequence with all values > thA). #'For t
#param th0 numeric; effect size under the null hypothesis
#param th0.seq monotone numeric vector of values getting closer to th0 from outside the
#th0-thA range (ie, if thA > th0, they should form an increasing sequence with all values < th0).
#The study will stop for futility at or before the k-th stage with probability \code{power.futility}.
#Its length should be equal to that of \code{thA.seq}. The default value of \code{NULL} implies no
#futility monitoring.
#param sig.level numeric; the one-sided significance level. The default is 0.025.
#param power numeric; the desired study power. The default is 0.9.
#param power.efficacy numeric; the probability of stopping for efficacy at stage k under \code{thA.seq}
#param power.futility numeric; the probability of stopping for futility at stage k under \code{th0.seq}
#param futility.type character string, one of \code{c("none", "non-binding", "binding")} or their
#unique abbreviations. Specifies whether a futility boundary should not be used at all ("none"), or if i
#is used, then whether the effect of the futility boundary should be included
#in the efficacy power/type I error calculations ("binding"), or not ("non-binding").
#param mix.theta numeric vector; specifies the set of alternatives under which the design is optimized.
# Defaults to \code{thA}.
#param mix.w numeric vector of length equal to that of \code{mix.theta}. The weights of the
#elements of \code{mix.theta} in the optimality criterion. It will be normalized to a sum of 1.
# Defaults to equal weights.
#param control list of parameters controlling the estimation algorithm to replace the default
#values returned by the \code{glsControl} function.
#return an object of class \code{gsDesignOC}
#author Aniko Szabo
#references Szabo, A, Tarima, S (?) Operating-characteristic guided design of group-sequential trials.
#keywords nonparametric design
#examples
#
#gsDesignOC(0.3, thA.seq = c(1, 0.5))
#
#name gsDesignOC
```

```
gsDesignOC <- function(thA, thA.seq, th0=0, th0.seq=NULL,
  sig.level = 0.025,
  power=0.9, power.efficacy=power, power.futility = power,
  futility.type=c("none","non-binding","binding"),
  mix.theta = thA,
  mix.w = rep(1, length(mix.theta)),
  control=list()){
  < Check inputs 6a >

  # create skeleton gsDesignOC object
  res <- list(thA=thA, thA.seq=thA.seq,
    th0=th0, th0.seq=th0.seq,
    sig.level = sig.level,
    power=power, power.efficacy = power.efficacy, power.futility = power.futility,
    futility.type=futility.type)
  class(res) <- "gsDesignOC"
```

```
< Define optimization function 6b >
```

```
k <- length(thA.seq) + 1
n.guess <- ztest.n(delta=c(thA.seq,thA)-th0, sd=1,
  sig.level=sig.level, power = power)
thA.seq <- c(thA.seq, thA)
```

*⟨ Check inputs 6a ⟩ ≡*

```

if (any(diff(c(thA.seq, thA)) * sign(th0-thA) <= 0))
  stop("'thA.seq' should approach thA in a monotone sequence outside the th0-thA range")

if (!is.null(th0.seq)){
  if (length(th0.seq) != length(thA.seq))
    stop("If specified, 'th0.seq' should have the same length as 'thA.seq'")
  if (any(diff(c(th0.seq, th0)) * sign(thA-th0) <= 0))
    stop("'th0.seq' should approach th0 in a monotone sequence outside the th0-thA range")
}

if (length(mix.w) != length(mix.theta))
  stop("'mix.w' should have the same length as 'mix.theta'")

if (power.efficacy > power)
  stop("The value of 'power.efficacy' should not exceed the value of 'power'.")

if (power.futility > 1-sig.level)
  stop("The value of 'power.futility' should not exceed the value of 1-'sig.level'.")

futility.type <- match.arg(futility.type)
if (futility.type != "none"){
  if (is.na(th0.seq)) stop("'th0.seq' should be specified if a futility bound is requested")
}

controlvals <- ocControl()
if (!missing(control))
  controlvals[names(control)] <- control

```

◇  
Fragment referenced in 5.

*⟨ Define optimization function 6b ⟩ ≡*

```

# n.vec is stage-specific sample sizes
.cp <- function(n.vec){

  n <- sum(n.vec)
  timing <- cumsum(n.vec/n)
  if (!is.null(sig.level_final)) timing <- head(timing, -1)

  res$n <- n
  res$timing <- timing
  res$bounds <- calc.bounds(res)
  dp <- oc(res, EN_theta=mix.theta, mix.w=mix.w)
  Q <-
    controlvals$optim.penalty * abs(dp$typeI - sig.level)/sqrt(sig.level*(1-sig.level)) +
    controlvals$optim.penalty * abs(dp$power - power)/sqrt(power*(1-power)) +
    dp$EN
  Q
}

```

◇  
Fragment referenced in 5.  
Uses: `calc.bounds` 8b, `oc` 7.

## 3. KEY SUPPORT FUNCTIONS

"../R/gsDesignOC.R" 7≡

```
#'Operating characteristics of a potential design
#'#'The values supplied in the function call replace the defaults and a list with all possible #'arguments
#'\code{gsDesignOC} function.
#'#'
#'\@export
#'\@param x object of class \code{gsDesignOC}
#'\@param EN_theta numeric vector of parameter values for which expected sample size calculation is performed
#'\@param mix.w numeric vector of positive weights for each value of \code{EN_theta}. Defaults to equal weights
#'\@return a list with the following elements:
#'\describe{
#'\item{typeI}{type I error, ie the probability of not crossing the efficacy boundary under theta=th0}
#'\item{power}{power, ie the probability of crossing the efficacy boundary under theta=thA}
#'\item{EN}{expected sample size under the requested alternatives}
#'}
#'\@author Aniko Szabo
#'\@keywords design
#'\@examples
#'#'
#'#'

oc <- function(x, EN_theta=x$thetaA, mix.w = rep(1, length(EN_theta))){

  if (length(mix.w) != length(EN_theta))
    stop("'EN_theta' and 'mix.w' should have the same length")
  if (!all(mix.w > 0))
    stop("'mix.w' should have only positive elements")

  # under H0
  res0 <- overall.crossprob(x, theta=x$theta0)
  # under Ha
  resA <- overall.crossprob(x, theta=x$thetaA)

  # under EN-alternatives
  en_vec <- numeric(length(EN_theta))
  for (i in seq_along(EN_theta)){
    th <- EN_theta[i]
    # check if we already calculated the EN at this value
    if (abs(th - x$theta0) < .Machine$double.eps){
      en_vec[i] <- res0$EN
    } else if (abs(th - x$thetaA) < .Machine$double.eps) {
      en_vec[i] <- resA$EN
    } else {
      en_vec[i] <- overall.crossprob(x, theta=th)
    }
  }
  en <- en_vec %*% mix.w / sum(mix.w)
  list(typeI = res0$cross.up,
       power = resA$cross.up,
       EN = en)
}
```

File defined by [5](#), [7](#), [8ab](#).

Defines: [oc](#) [6b](#).

```
"../R/gsDesignOC.R" 8a≡
```

```
#'Control values for gsDesignOC
#'
#'#The values supplied in the function call replace the defaults and a list with all possible
#'#arguments is returned. The returned list is used as the \code{control} argument to the
#'#\code{gsDesignOC} function.
#'
#'#@export
#'#@param optim.penalty numeric; relative weight of terms ensuring target type I and type II
#'#errors versus sample size in optimization routine
#'#@return a list with components for each of the possible arguments
#'#@author Aniko Szabo
#'#@keywords design
#'#@examples
#'#
#'#ocControl(optim.penalty = 100)

ocControl <- function(optim.penalty = 1000){
  list(optim.penalty = optim.penalty)
}
◇
```

File defined by [5](#), [7](#), [8ab](#).

#### 4. BOUNDARY ACHIEVING THE DESIRED OPERATING CHARACTERISTICS

Given the alpha-spending sequence and using the probability targets defined in [Table 2](#), we can derive the information times  $\mathcal{I}_k$ , and boundaries  $u_k$  and  $l_k$  using the recursive construction process described in the proof of [Theorem ??](#).

```
"../R/gsDesignOC.R" 8b≡
```

```
#'#Calculate the information times and efficacy/futility boundary values given the alpha-spending sequence

#'#@export
#'#@param x an object of class \code{gsDesignOC}

calc.bounds <- function(x, alpha.seq){

  n.stages <- length(x$thA.seq) + 1
  alpha.cum <- cumsum(alpha.seq)
  ub <- rep(20, n.stages)
  lb <- rep(-20, n.stages)
  ivec <- rep(NA, n.stages)

  < Define exceedance probability functions 9b >
  < Construct first stage 9a >

  if (n.stages > 1){
    for (k in 2:n.stages){
      < Construct next stage 10b >
    }
  }

  if (x$futility.type == "non-binding"){
    < Add non-binding lower bound 11d >
  }

  x$upper <- ub
```



```

x$lower <- lb
x$info <- ivec
return(x)
}

```

◇

File defined by 5, 7, 8ab.

Defines: calc.bounds 6b.

The first stage is based on the fixed sample-size formula for alternative  $\theta_{A1}$ , power  $\pi_E$ , and significance level  $\Delta\alpha_1$ .

⟨ Construct first stage 9a ⟩ ≡

```

ivec[1] <- ztest.I(delta=x$thA.seq[1]-x$th0, sig.level=alpha.seq[1], power=x$power.efficacy)
ub[1] <- qnorm(alpha.seq[1], lower=FALSE)

```

◇

Fragment referenced in 8b.

Uses: ztest.I 12a.

The cumulative rejection / acceptance probabilities depend on the type of the futility boundary.

⟨ Define exceedance probability functions 9b ⟩ ≡

```

⟨ Define upper bound exceedance 9c ⟩
⟨ Define function to find u(I) 10a ⟩
⟨ Define lower bound exceedance 11a ⟩
⟨ Define function to find l 11b ⟩

```

◇

Fragment referenced in 8b.

**4.1. Upper bound exceedance.** The functions  $a$  and  $b$  in the proof of Theorem ?? have a similar form, so we will define the following function:

$$e(u | \mathcal{I}) = Pr\left(\bigcup_{i=1}^{k-1} \{l_1 < Z_1 < u_1, \dots, l_{i-1} < Z_{i-1} < u_{i-1}, Z_i \geq u_i\} \bigcup \{l_1 < Z_1 < u_1, \dots, l_{k-1} < Z_{k-1} < u_{k-1}, Z(\mathcal{I}) \geq u\} \mid \theta\right).$$

If  $l_i$  is set to  $-\infty$ , then that is equivalent to not having a lower boundary. We will need to solve equations of the form  $e(u) = c$ , so we will actually define a function for the difference of the LHS and RHS.

⟨ Define upper bound exceedance 9c ⟩ ≡

```

exc <- function(u, I, stage, theta, target){
  gg <- gsDesign::gsProbability(k=stage,
                                theta=theta,
                                n.I = c(ivec[1:(stage-1)], I),
                                a = c(lb[1:(stage-1)], -20),
                                b = c(ub[1:(stage-1)], u))
  sum(gg$upper$prob[1:stage]) - target
}

```

◇

Fragment referenced in 9b.

To find  $u(I)$  we need to solve  $a(u|\mathcal{I}) = \sum_{i=1}^k \Delta\alpha_i = \tilde{\alpha}_k$ . Since  $a(z_\alpha | \mathcal{I}_N) \geq \alpha$  and  $a(\infty|\mathcal{I}_N) = \tilde{\alpha}_{k-1}$ , the solution for fixed  $\mathcal{I}_N$  is between  $z_\alpha$  and 20 (which is essentially  $\infty$ ).

*⟨ Define function to find  $u(I)$  10a ⟩*  $\equiv$

```
uI <- function(I, stage){
  res <- uniroot(exc, interval=c(qnorm(x$sig.level, lower=FALSE), 20),
    I = I, stage=stage, theta=x$th0, target = alpha.cum[stage],
    extendInt = "downX")
  res$root
}
```

Fragment referenced in [9b](#).

#### 4.2. Construct next stage.

*⟨ Construct next stage 10b ⟩*  $\equiv$

```
if (x$futility.type == "binding"){
  ⟨ Find lower bound for previous stage 11c ⟩
}

⟨ Find I for next stage 10c ⟩
```

Fragment referenced in [8b](#).

To find  $I_k$ , we need to solve  $b(I) = \pi_E$ , if  $k < K$  and  $= \pi$  for  $k = K$ . Note that  $b(I_{k-1}) \leq \pi_E$  and  $b(\mathcal{I}_{fix}(\theta_{Ak} - \theta_0, \tilde{\alpha}_k, \pi_E)) \leq \pi_E$ , we can start the search at the maximum of these two values.

*⟨ Find I for next stage 10c ⟩*  $\equiv$

```
if (k == n.stages){
  power.target <- x$power
  .th <- x$thA
} else {
  power.target <- x$power.efficacy
  .th <- x$thA.seq[k]
}

minI <- max(ztest.I(delta = .th - x$th0, power=power.target, sig.level=alpha.cum[k]),
  ivec[k-1])
resI <- uniroot(function(ii)exc(uI(ii, k), ii, stage=k, theta=.th,
  target = power.target),
  interval=c(minI, 2*minI),
  extendInt = "upX")
ivec[k] <- resI$root
ub[k] <- lb[k] <- uI(resI$root, k)
```

Fragment referenced in [10b](#).

Uses: `ztest.I` [12a](#).

**4.3. Add lower bound.** The lower bound will be computed only when the information and upper bound for a stage has been determined.

*⟨ Define lower bound exceedance 11a ⟩ ≡*

```
exc_low <- function(l, stage, theta, target){
  gg <- gsDesign::gsProbability(k=stage,
                                theta=theta,
                                n.I = ivec[1:stage],
                                a = c(head(lb, stage-1), 1),
                                b = ub[1:stage])
  sum(gg$lower$prob[1:stage]) - target
}
```

Fragment referenced in [9b](#).

*⟨ Define function to find l 11b ⟩ ≡*

```
lI <- function(stage, theta){
  res <- uniroot(exc_low, interval=c(-20, qnorm(x$sig.level, lower=FALSE)),
                stage=stage, theta=theta, target = x$power.futility,
                extendInt = "upX")
  res$root
}
```

Fragment referenced in [9b](#).

We need to find the lower bound for stage  $k - 1$ , before computing the size of the next stage.

*⟨ Find lower bound for previous stage 11c ⟩ ≡*

```
lb[k-1] <- lI(stage=k-1, theta=x$th0.seq[k-1])
```

Fragment referenced in [10b](#).

For a non-binding boundary, the lower bound is added only after the entire upper bound has been calculated.

*⟨ Add non-binding lower bound 11d ⟩ ≡*

```
for (k in 1:(n.stages-1)){
  lb[k] <- lI(stage=k, theta=x$th0.seq[k])
}
```

Fragment referenced in [8b](#).

## 5. UTILITY HELP-FUNCTIONS

"../R/Utility.R" 12a≡

```
#'Fixed sample-size information for one-sided test
#
#'The \code{ztest.I} function finds the information required of a one-sided Z-test with given power
#'and significance level
#
#'@param delta numeric; targeted standardized effect size
#'@param sig.level numeric; one-sided significance level
#'@param power numeric; target power
#'@return numeric value with the (fractional) information achieving the target power
#'@author Aniko Szabo
#'@keywords internal
#'@examples
#
#'ztest.I(delta=1, sig.level=0.05, power=0.9)

ztest.I <- function(delta, sig.level, power){
  za <- qnorm(sig.level, lower=FALSE)
  zb <- qnorm(power)
  I <- (za+zb)^2 /delta^2
  I
}
◇
```

File defined by [12ab](#).

Defines: `ztest.I` [9a](#), [10c](#).

"../R/Utility.R" 12b≡

```
#'Conversion between boundary and nominal significance level
#
#'The \code{nom.to.bnd} and \code{bnd.to.nom} functions perform conversion between the boundary and
#'matching significance level.
#
#'@param nominal.level numeric vector or matrix with two rows of the nominal significance level for each
#' When a matrix, row 1 corresponds to the efficacy boundary and row 2 to the futility. Defaults to NULL
#'@param cutoffs numeric vector or numeric matrix with two rows of the z-test cutoffs for each stage.
#' When a matrix, row 1 corresponds to the efficacy boundary and row 2 to the futility. Defaults to NULL
#'@param n integer vector of sample sizes of each stage. Its sum is the total study sample size.
#'@param theta.null numeric, the null hypothesis being tested
#'@param theta.alt numeric, the alternative hypothesis being tested for the futility bound calculation
#'@keywords internal
#'@examples
#
#'## only efficacy
#'(b1 <- nom.to.bnd(nominal.level = c(0.01, 0.02), n = c(30, 50)))
#'bnd.to.nom(b1, n=c(30,50))

nom.to.bnd <- function(nominal.level, n, theta.null, theta.alt=NULL){
  nominal.level <- rbind(nominal.level) # raises to matrix if one row
  m <- nrow(nominal.level)
  k <- ncol(nominal.level)
  cutoffs <- matrix(NA, nrow=m, ncol=k)
  cutoffs[1, ] <- qnorm(nominal.level[1,], lower=FALSE)
  if (k > 1) {
    # flip direction + shift hypothesis
    cutoffs[2, ] <- qnorm(nominal.level[2,], lower=TRUE) + sqrt(n)*(theta.alt -theta.null)
  }
}
```

```

    }
    cutoffs[1:k,] # drops to vector if one row
}

#'@describeIn nom.to.bnd Conversion from nominal significance level to boundary
#'@export
#'@keywords internal
#'@examples
#'## both efficacy and futility
#'(b2 <- nom.to.bnd(nominal.level = rbind(c(0.01, 0.02), c(0.05,0.1)),
#',
#', n = c(30, 50)))
#',bnd.to.nom(b2, n=c(30,50))

bnd.to.nom <- function(cutoffs, n, theta.null, theta.alt=NULL){
  cutoffs <- rbind(cutoffs) # raises to matrix if one row
  m <- nrow(cutoffs)
  k <- ncol(cutoffs)
  noms <- matrix(NA, nrow=m, ncol=k)
  noms[1,] <- pnorm(cutoffs[1,], lower=FALSE)
  if (k > 1) {
    # flip direction + shift hypothesis
    noms[2,] <- pnorm(cutoffs[2,] - sqrt(n)*(theta.alt-theta.null), lower=TRUE)
  }
  noms[1:k,] # drops to vector if one row
}

```

◇  
File defined by [12ab](#).