### 5.3. Gumbel's and Joe's polynomial

For computing the log-likelihood for the Archimedean Gumbel or Joe copula, we need an efficient way of evaluating the logarithm of the density $c_{\theta}(\boldsymbol{u})$ as given in Section 3.4, Parts 4 and 5, respectively. The challenge is to evaluate the logarithm of the polynomials involved. For this the following auxiliary results are essential. Their proofs are straightforward and thus omitted.

## Lemma 5.1

1. Let $x_{i} \geq 0, i \in\{1, \ldots, n\}$, such that $\sum_{i=1}^{n} x_{i}>0$. Furthermore, let $b_{i}=\log x_{i}, i \in\{1, \ldots, n\}$, with $\log 0=-\infty$, and let $b_{\max }=\max _{1 \leq i \leq n} b_{i}$. Then

$$
\begin{equation*}
\log \sum_{i=1}^{n} x_{i}=b_{\max }+\log \sum_{i=1}^{n} \exp \left(b_{i}-b_{\max }\right) \tag{14}
\end{equation*}
$$

2. Let $x_{i} \in \mathbb{R}, i \in\{1, \ldots, n\}$, such that $\sum_{i=1}^{n} x_{i}>0$. Furthermore, let $s_{i}=\operatorname{sign} x_{i}, b_{i}=\log \left|x_{i}\right|$, $i \in\{1, \ldots, n\}$, with $\log 0=-\infty$ and let $b_{\max }=\max _{1 \leq i \leq n} b_{i}$. Then

$$
\begin{equation*}
\log \sum_{i=1}^{n} x_{i}=b_{\max }+\log \left(\sum_{\substack{i=1 \\ i: s_{i}=1}}^{n} \exp \left(b_{(i)}-b_{\max }\right)-\sum_{\substack{i=1 \\ i: s_{i}=-1}}^{n} \exp \left(b_{(i)}-b_{\max }\right)\right) \tag{15}
\end{equation*}
$$

where $b_{(i)}$ denotes the $i$ th smallest value of $b_{i}, i \in\{1, \ldots, n\}$.
The ideas behind Lemma 5.11 and 2 are implemented in the (non-exported) functions lsum and lssum in the R package copula.

Although mathematically straightforward, Lemma 5.1 has an important consequence for evaluating logarithms of polynomials such as $P_{d, \alpha}^{\mathrm{G}}$ for Gumbel's or $P_{d, \alpha}^{\mathrm{J}}$ for Joe's density. Depending on the evaluation point, it might happen that the value of the polynomial is not representable in computer arithmetic and thus one cannot first compute the value of the polynomial and take the logarithm afterwards. Instead, Formula (14) suggests a "intelligent" (numerically stable) logarithm to compute such polynomials (or sums). By taking out the maximum of the $b_{i}$, it is guaranteed that the exponentials which are summed up are all in $[0,1]$ and thus the sum takes on values in $[0, n]$, representable in computer arithmetic. This trick solves the numerical issues for computing $P_{d, \alpha}^{\mathrm{J}}$ and thus for computing the log-likelihood of a Joe copula. The evaluation of $P_{d, \alpha}^{\mathrm{J}}$ is implemented as (non-exported) function polyJ in the R package copula. It is called with default method log.poly implementing the trick described above when evaluating the density of a Joe copula via the slot dacopula; two other, less efficient methods are also available, one of which is a straightforward polynomial evaluation (poly).

Formula (15) takes the above idea of an intelligent logarithm a step further, by dealing with possibly negative summands. The summands in each sum are ordered in increasing order to prevent cancellation. This formula is helpful in computing $P_{d, \alpha}^{\mathrm{G}}$. However, the situation turns out to be more challenging for Gumbel's family. All in all, several different methods for the evaluation of $P_{d, \alpha}^{\mathrm{G}}$ were implemented. They are based on the following results about $P_{d, \alpha}^{\mathrm{G}}$ and described below, where here and in the following, $\alpha=1 / \theta \in(0,1]$.

